Statistical Techniques in Robotics (16-831, S20) Lecture #04 (Wednesday, February 10)

PWEA (WMA, RWMA)

Lecturer: Kris Kitani   *Scribes: Alex Withers, Roshni Kaushik, and Cornelia Bauer*

1 Review

1.1 Realizability

In previous lectures one of our fundamental assumptions was that each of our hypothesis sets were realizable. If a hypothesis set is realizable, it means that at least one mapping of observations to outcomes is perfect and makes no mistakes. This assumption was useful for learning and understanding the greedy and halving algorithms, and as an introduction to online learning, but is rarely applicable in practice. This lecture and the coming lectures will not assume realizability, and will not assume that there exists a perfect hypothesis.

1.2 Regret

Previously, we learned that regret, in this context, is the cumulative loss of the learning algorithm minus the cumulative loss of the best hypothesis in the hypothesis set. This is represented in Equation 1, where $(\sum_{t=1}^{T} l(\hat{y}^{(t)}, y^{(t)}))$ represents the cumulative loss of the learning algorithm and $(\min_{h \in H} \sum_{t=1}^{T} l(h(x^{(t)}), y^{(t)}))$ represents the cumulative loss of the best hypothesis in the hypothesis set.

$$R^{(T)}(H) = \sum_{t=1}^{T} l(\hat{y}^{(t)}, y^{(t)}) - \min_{h \in H} \sum_{t=1}^{T} l(h(x^{(t)}), y^{(t)}) \tag{1}$$

Regret is used to define bounds on the performance of online learning algorithms that do not follow the realizability assumption. Minimizing the upper bound of the regret will improve the performance of the online learning algorithm, with the hope of achieving a **no-regret** algorithm. If the average regret approaches 0 as $T \to \infty$, the algorithm is considered no-regret. A no-regret algorithm will perform no worse than the best hypothesis in hindsight. If the upper bound of the regret grows sub-linearly as T approaches infinity, then the algorithm is considered to be a no-regret algorithm.

1.3 Halving (Majority) Algorithm

The Halving or Majority algorithm was introduced last time, and while it relied on the realizability assumption, it is good to review the important parts of this algorithm. The idea is to remove at least half of the version space if the learner makes a mistake. This is possible because the learner follows a hypothesis that is a part of the majority choice, and when the majority is wrong, they are all eliminated. This selection of the hypothesis decreases the mistake bound to $M_{halving}(\mathcal{H}) \leq \log_2 |\mathcal{H}|$ from $M_{greedy}(H) \leq |\mathcal{H}| - 1$ in the greedy algorithm. This is a significant increase in performance, but it could still be better.

1.4 Randomized Greedy Algorithm

Randomization of the greedy algorithm is expected to provide even better performance than the Halving algorithm. By randomizing the choice of hypothesis from the version space, the random seed gives an advantage over an adversarial nature and reduces mistakes in the worst case scenarios, which is the upper bound of mistakes made by the greedy algorithm. Thus the **expected** mistake bound is $M_{randGreedy} \leq \ln(|\mathcal{H}|)$.

2 Weighted Majority Algorithm (WMA)

The Weighted Majority Algorithm (WMA) was introduced in [2] and allowed for a reduction in confidence experts as they make mistakes, without eliminating them as in the Greedy Algorithm. This algorithm introduces a set of weights (w_n) that change at every iteration t. The learner predicts $\hat{y}$ based on the weighted combination of the experts' predictions. Experts that make an incorrect prediction will have their weights reduced using discount factor η. Additional reading on WMA and its connections to other statistical techniques can be found in [1].

Algorithm 1 Weighted Majority Algorithm (WMA)

```
$\mathbf{w}^{(1)} \leftarrow \{w_n^{(1)} = 1\}_{n=1}^N$                              ▷ Initialize weights
$\eta \leq \frac{1}{2}$                                                         ▷ Initialize penalty parameter
**for** $t = 1, \cdots, T$ **do**
    RECEIVE $(\mathbf{x}^{(t)} \in \{-1, 1\}^N)$                                ▷ Receive expert prediction
    $\hat{y}^{(t)} = \text{sign}\Big(\sum_{n=1}^N x_n^{(t)} \cdot w_n^{(t)}\Big) \in \{-1, 1\}$   ▷ Make learner prediction
    RECEIVE $(y^{(t)} \in \{-1, 1\})$                                           ▷ Receive true outcome
    $w_n^{(t+1)} \leftarrow w_n^{(t)}\big(1 - \eta \cdot \mathbf{1}[y^{(t)} \neq x_n^{(t)}]\big)$   ▷ Update weights based on penalty
**end for**
```

Let us derive the relative mistake bound and regret bound for WMA. First, let us review some notation.

- $M^{(T)}$: the number of mistakes made by the learner at iteration T
- N: the number of experts
- $m_n^{(T)}$: the number of mistakes using hypothesis h at iteration T
- h: a hypothesis, which for WMA is one possible weighted combination of experts

We will also use these two identities:

Lemma 1. *For $0 < x < 1$, following two inequalities hold:*

$$-x - x^2 \leq \ln(1 - x) \tag{2}$$

$$\ln(1 - x) \leq -x \tag{3}$$

Proof. The proof for these can derived from the Taylor series expansion of $\ln(1+x)$

Theorem 2. *(Mistake bound of WMA) $M^{(t)}$*

$$M^{(t)} \leq 2(1+\eta)m_i^{(t)} + \frac{2\ln N}{\eta}$$

Proof. Let us first define the potential function $\Phi^{(t)}$ as:

$$\Phi^{(t)} = \sum_{n=1}^{N} w_n^{(t)} \tag{4}$$

The WMA learning rule implies that at least half of the weighted experts are mistaken when an incorrect prediction $\hat{y}^{(t)}$ is made at iteration t.

From the weight update rule, we know that all experts who make a mistake will have their weights reduced by $(1-\eta)$.

The potential at the iteration after a mistake will be:

$$\Phi^{(t+1)} \leq \Phi^{(t)}\Big(\frac{1}{2} + \frac{1}{2}(1-\eta)\Big) = \Phi^{(t)}\Big(1 - \frac{\eta}{2}\Big)$$

By induction, we can compute this **upper bound** in terms of $\Phi^{(1)}$:

$$\Phi^{(t+1)} \leq \Phi^{(1)}\Big(1 - \frac{\eta}{2}\Big)^{M^{(t)}} = N\Big(1 - \frac{\eta}{2}\Big)^{M^{(t)}}$$

To calculate the lower bound, we can first rewrite Line 7 from the algorithm (the weight update step) in terms of our desired variables.

$$w_i^{(t+1)} = (1-\eta)^{m_i^{(t)}}$$

We go back to the definition of the potential (Equation 4) as the sum of the weights. A trivial lower bound on a sum is a single element of that sum or:

$$w_i^{(t+1)} \leq \sum_{n=1}^{N} w_n^{(t+1)} = \Phi^{(t+1)}$$

Combining these together we have our **lower bound** as:

$$(1-\eta)^{m_i^{(t)}} \leq \Phi^{(t+1)}$$

Let us combine the upper and lower bounds by squeezing our potential function:

$$(1-\eta)^{m_i^{(t)}} \leq N\Big(1 - \frac{\eta}{2}\Big)^{M^{(t)}}$$

Taking the natural log of both sides preserves the inequality.

$$m_i^{(t)} \ln(1-\eta) \leq \ln N + M^{(t)} \ln\Big(1 - \frac{\eta}{2}\Big)$$

Using Equation 2, we can simplify the left hand side:

$$m_i^{(t)}(-\eta - \eta^2) \leq \ln N + M^{(t)} \ln\Big(1 - \frac{\eta}{2}\Big)$$

Using Equation 3, we can simplify the right hand side:

$$-m_i^{(t)}(\eta + \eta^2) \leq \ln N + M^{(t)}\Big(-\frac{\eta}{2}\Big)$$

Rewriting, we can derive the mistake bound of WMA

$$M^{(t)}\Big(\frac{\eta}{2}\Big) \leq \ln N + m_i^{(t)}(\eta + \eta^2)$$

$$M^{(t)} \leq \Big(\frac{2}{\eta}\Big) \ln N + \Big(\frac{2}{\eta}\Big) m_i^{(t)}(\eta + \eta^2)$$

$$M^{(t)} \leq \frac{2 \ln N}{\eta} + 2m_i^{(t)}(1 + \eta)$$

□

The mistake bound of WMA depends on the number of mistakes an expert makes, so this is a **relative** mistake bound. To compute the **regret bound** of WMA, we rearrange terms:

$$R(h_i) = M^{(T)} - m_i^{(T)} \leq \frac{2 \ln N}{\eta} + m_i^{(T)} + 2m_i^{(T)}\eta$$

Theorem 3. *WMA is a* ***bounded regret*** *algorithm*

Proof. The first term of the regret is $O(1)$ and the second and third terms of the regret are $O(T)$. The average regret ($\frac{R(h_i)}{T}$), therefore, is $O(1)$ and approaches a constant as $T \to \infty$. This means WMA is a bounded regret algorithm. □

WMA is not a no-regret algorithm as proposed, but as we will see in the next section, adding randomization will allow us to reduce the number of mistakes and get a no-regret algorithm.

3 Randomized Weighted Majority Algorithm (RWMA)

The Randomized Weighted Majority Algorithm (RWMA) improves the mistake bound by introducing randomization. This algorithm can also be found under the name *Generalized Weighted Majority*, *Exponentiated Weighted Majority*, or *Hedge Algorithm*.

Recall that for WMA, the learner predicts $\hat{y}$ based on the weighted combination of the experts' advice:

$$\hat{y}^{(t)} = \operatorname{sign}\Big(\sum_{n=1}^{N} x_n^{(t)} \cdot w_n^{(t)}\Big) \in \{-1, 1\}$$

For the **randomized** variant, RWMA, we instead define a selector function h, given by the multinomial distribution over the relative expert weights $w^{(t)}$

$$h \sim \textsc{Multinomial}(\boldsymbol{w}^{(t)}/\Phi^{(t)})$$

where $\Phi^{(t)}$ is defined in Equation 4.

The learner's prediction is now sampled from the selector function

$$\hat{y}^{(t)} = h(\boldsymbol{x}^{(t)})$$

and the weights are updated based on discrete loss.

Algorithm 2 Randomized Weighted Majority Algorithm (RWMA)

```
1: $\mathbf{w}^{(1)} \leftarrow \{w_n^{(1)} = 1\}_{n=1}^N$    ▷ Initialize weights
2: $\eta \leq \frac{1}{2}$    ▷ Initialize penalty parameter
3: for $t = 1, \cdots, T$ do
4:     Receive $(\boldsymbol{x}^{(t)} \in \{-1, 1\}^N)$    ▷ Receive expert prediction
5:     $h \sim \textsc{Multinomial}(\boldsymbol{w}^{(t)}/\Phi^{(t)})$    ▷ Selector function
6:     $\hat{y}^{(t)} = h(\boldsymbol{x}^{(t)})$    ▷ Make learner prediction
7:     Receive $(y^{(t)} \in \{-1, 1\})$    ▷ Receive true outcome
8:     $w_n^{(t+1)} \leftarrow w_n^{(t)}\big(1 - \eta \cdot \mathbf{1}[y^{(t)} \neq h_n(\boldsymbol{x})^{(t)}]\big)$    ▷ Update weights based on penalty
9: end for
```

We will now derive the mistake bound for RWMA, and show that randomization improves the mistake bound by a factor of 2, compared to WMA.

For this derivation, we require the following lemma:

Lemma 4. $e^x \geq 1 + x \quad \forall x \in \mathbb{R}$

Proof. This can be derived from the Taylor series expansion of e^x.

Theorem 5. *(Mistake bound of RWMA) Given $M^{(t)}$, the number of mistakes by the learner until iteration t, and $m_n^{(t)}$ the number of mistakes made by the n-th hypothesis until t, N the number of experts and η the penalty rate. Then, the expected number of mistakes by the learner $E[M^{(T)}]$ is upper-bounded by:*

$$E[M^{(T)}] \leq (1 + \eta)m_n^{(T)} + \frac{\ln N}{\eta}$$

Proof. Given the weight update rule given in Line 8 of Algorithm 2, the potential function at iteration $t + 1$ can be written as

$$\Phi^{(t+1)} = \sum_n w_n^{(t)}(1 - \eta\alpha_n^{(t)}) \qquad (\text{where } \alpha_i^{(t)} = \mathbf{1}[y^{(t)} \neq y_i^{(t)}]) \tag{5}$$

Equation 5 can be rewritten as

$$
\begin{aligned}
\Phi^{(t+1)} &= \sum_n w_n^{(t)}(1 - \eta\alpha_n^{(t)}) \\
&= \sum_n w_n^{(t)} - \sum_n \eta w_n^{(t)} \alpha_n^{(t)} \\
&= \Phi^{(t)} - \frac{\Phi^{(t)}}{\Phi^{(t)}} \sum_n \eta w_n^{(t)} \alpha_n^{(t)} \\
&= \Phi^{(t)} - \Phi^{(t)} \eta \sum_n \frac{w_n^{(t)}}{\Phi^{(t)}} \alpha_n^{(t)} \\
&= \Phi^{(t)} - \Phi^{(t)} \eta \sum_n p_n^{(t)} \alpha_n^{(t)} \quad (\text{where } p_n^{(t)} = \frac{w_n^{(t)}}{\Phi^{(t)}}) \\
&= \Phi^{(t)}(1 - \eta \sum_n \alpha_n^{(t)} p_n^{(t)}) \qquad (6)
\end{aligned}
$$

By induction, we find

$$\Phi^{(T)} = \Phi^{(1)} \prod_{t=1}^{T-1} (1 - \eta \sum_n p_n^{(t)} \alpha_n^{(t)}) \qquad (7)$$

We can now make use of the approximation given by Lemma 4, and $\Phi^{(1)} = N$ to derive the **upper bound**:

$$
\begin{aligned}
\Phi^{(T+1)} &\leq N \prod_{t=1}^{T} \exp(-\eta \sum_n p_n^{(t)} \alpha_n^{(t)}) \\
&= N \exp(-\eta \sum_{t=1}^{T} E_p[\mathbf{1}[y^{(t)} \neq \hat{y}_n^{(t)}]) \;\; (\text{where } \sum_n p_n^{(t)} \alpha_n^{(t)} = E_p[\mathbf{1}[y^{(t)} \neq \hat{y}_n^{(t)}]) \qquad (8) \\
&= N \exp(-\eta E[M^{(T)}]) \qquad (9)
\end{aligned}
$$

Regarding the **lower bound**, we know that the trivial lower bound of a sum is one of its elements, which leads us to

$$w_n^{(t+1)} \leq \Phi^{(t+1)}$$

Given the weight update rule $w_n^{(t+1)} = (1 - \eta)^{m_n^{(t)}}$, we can rewrite this to

$$(1 - \eta)^{m_n^{(t)}} \leq \Phi^{(t+1)}$$

where $m_n^{(t)}$ denotes the number of mistakes of expert n at iterration t. We can now combine the upper and lower bound on the potential function,

$$(1 - \eta)^{m_n^{(T)}} \leq N \exp(-\eta E[M^{(T)}])$$

Applying the natural log, we rewrite this to

$$m_n^{(T)} \ln(1-\eta) \leq \ln N - \eta E[M^{(T)}]$$

Using Equation 2 from Lemma 1, we simplify the left hand side:

$$m_n^{(T)}(-\eta - \eta^2) \leq \ln N - \eta E[M^{(T)}]$$

$$-m_n^{(T)}(1+\eta) \leq \frac{\ln N}{\eta} - E[M^{(T)}]$$

We can rewrite this to

$$E[M^{(T)}] \leq \frac{\ln N}{\eta} + m_n^{(T)}(1+\eta)$$

and we have obtained the mistake bound of RWMA. □

To calculate the **regret bound of RWMA**, we can rearrange terms:

$$E[R] = E[M^{(T)}] - m_n^{(T)} \leq \frac{\ln N}{\eta} + \eta m_n^{(T)}$$

Theorem 6. *RWMA is a* ***no-regret*** *algorithm*

Proof. If we examine the terms in Equation 5, we find that $m_n^{(T)}$ is $O(T)$, and $\ln(T)$ is $O(1)$. If we set parameter η as

$$\eta^{(t)} = \frac{1}{\sqrt{t}}$$

we find that both terms on the right hand side are now $O(\sqrt{T})$, which means that regret grows sub-linearly over time. RWMA is therefore a no-regret algorithm. □

References

[1] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[2] N. Littlestone, M. K. Warmuth, et al. *The weighted majority algorithm.*